



PATENT

- 1 -

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of:	:	Before the Examiner:
Stoodley et al.	:	Tang, Kuo Liang J.
	:	
Serial No.: 09/872,586	:	Group Art Unit: 2122
	:	
Filing Date: June 1, 2001	:	
	:	IBM Corporation
Title: ADAPTING POLYMORPHIC:	:	P.O. Box 12195
INLINE CACHES FOR	:	Dept. 9CCA, Bldg. 002-2
MULTITHREADED	:	Research Triangle Park, NC 27709
COMPUTING	:	

APPEAL BRIEF


Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. **REAL PARTY IN INTEREST**

The real party in interest is International Business Machines, Inc., which is the assignee of the entire right, title and interest in the above-identified patent application.

CERTIFICATION UNDER 37 C.F.R. §1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450, on April 20, 2005.



Signature

04/25/2005 EFLORES 00000051 090461 09872586

01 FC:1402 500.00 DA

Serena Beller
(Printed name of person certifying)

II. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants, Appellants' legal representative or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-30 are pending in the Application. Claims 1-30 stand rejected. Claims 1-30 are appealed.

IV. STATUS OF AMENDMENTS

The Appellants' response to the Office Action having a mailing date of April 5, 2004, has been considered, but the Examiner indicated that it did not place the application in condition for allowance because Appellants' arguments were deemed unpersuasive.

V. SUMMARY OF CLAIMED SUBJECT MATTER

In one embodiment of the present invention, a method for implementing a polymorphic call site in a computer system executing an object oriented program may comprise the step of creating a template of a polymorphic inline cache for a polymorphic call site, the template having a plurality of slots. Specification, page 7, lines 14-17; Specification, page 14, line 30 – page 15, line 19; Specification, page 24, claim 1, lines 1-4; Figure 2, element 124. The method may further comprise executing the polymorphic inline cache with an object of type k. Specification, page 7, line 18; Specification, page 15, lines 21-22; Specification, page 24, claim 1, line 5. The method may further comprise invoking a polymorphic inline cache initialisation routine. Specification, page 7, lines 18-19; Specification, page 15, lines 22-28; Specification, page 24, claim 1, line 6; Figure 2, step 210. The method may further

comprise finding an available k slot of the polymorphic inline cache. Specification, page 7, line 19; Specification, page 15, lines 22-28; Specification, page 24, claim 1, line 7; Figure 2, step 210. The method may further comprise locking the k slot of the polymorphic inline cache. Specification, page 7, line 20; Specification, page 15, lines 28-30; Specification, page 24, claim 1, line 8; Figure 2, step 210. The method may further comprise searching for a k method to call for the object of type k. Specification, page 7, lines 20-21; Specification, page 16, lines 13-14; Specification, page 24, claim 1, line 9; Figure 2, step 220. The method may further comprise filling the k slot with a call instruction to the k method. Specification, page 7, line 21; Specification, page 16, lines 14-17; Specification, page 24, claim 1, line 10; Figure 2, step 240. The method may further comprise unlocking the k slot to complete the k slot. Specification, page 7, lines 21-22; Specification, page 16, lines 19-26; Specification, page 24, claim 1, line 11; Figure 2, step 260. The method may further comprise calling the k method of the object of type k. Specification, page 7, line 22; Specification, page 16, lines 26-28; Specification, page 24, claim 1, line 12; Figure 2, step 270.

In another embodiment of the present invention, a system in a computing environment for implementing a polymorphic call sit may comprise a central processing unit for executing an application. Specification, page 9, lines 10-11; Specification, page 12, lines 3-13; Specification, page 26, claim 19, lines 1-4; Figure 1, elements 101, 122. The system may further comprise a memory connected to the central processing unit via a bus. Specification, page 9, lines 11-12; Specification, page 12, lines 3-19; Specification, page 26, claim 19, line 6; Figure 1, elements 101, 102, 130. The system may further comprise at least one input/output device connected to the bus and connected to a network interface to an external computer network. Specification, page 9, lines 12-14; Specification, page 12, line 25 – page 13, line 22; Specification, page 26, claim 19, lines 8-9; Figure 1, elements 134, 138, 140, 148. The system may further comprise a just-in-time compiler for compiling object

oriented applications for execution. Specification, page 9, lines 14-15; Specification, page 12, lines 17-19; Specification, page 14, line 30- page 15, line 19; Specification, page 26, claim 19, line 11. The system may further comprise a polymorphic inline cache created by the just-in-time compiler, the polymorphic inline cache implementing a lockable slot for each individual object type to a polymorphic call site in the application. Specification, page 9, lines 15-17; Specification, page 14, line 30 – page 16, line 11; Specification, page 26, claim 19, lines 12-14; Figure 2, step 210.

In another embodiment of the present invention, a system in a computing environment for implementing a polymorphic call site may comprise a central processing unit for executing an application. Specification, page 9, lines 19-20; Specification, page 12, lines 3-13; Specification, page 26, claim 20, lines 1-2; Figure 1, elements 101, 122. The system may further comprise a memory connected to the central processing unit via a bus. Specification, page 9, lines 20-21; Specification, page 12, lines 3-19; Specification, page 27, claim 20, line 3; Figure 1, elements 101, 102, 130. The system may further comprise a network interface to connect to an external computer network. Specification, page 9, line 21; Specification, page 13, lines 11-12; Specification, page 27, claim 20, line 4; Figure 1, element 160. The system may further comprise a polymorphic inline cache having a plurality of slots, each slot allocated to an object type of a method and locked to other object types of a polymorphic call site in the application. Specification, page 9, lines 21-23; Specification, page 14, line 30 – page 15, line 30; Specification, page 27, claim 20, lines 5-7; Figure 2, step 210.

In another embodiment of the present invention, a method of using a polymorphic inline cache may comprise the step of calling a first method having a first object type from an executing object oriented program. Specification, page 10, lines 9-11; Specification, page 14, line 30 – page 15, line 19; Specification, page 27, claim 21, lines 1-3. The method may further comprise locking a first slot of the

polymorphic inline cache with a call to the first method of the first object type. Specification, page 10, lines 11-12; Specification, page 14, line 30 – page 16, line 11; Specification, page 27, claim 21, lines 4-5; Figure 2, step 210. The method may further comprise executing the first slot of the polymorphic inline cache. Specification, page 10, line 12; Specification, page 14, line 30 – page 16, line 27; Specification, page 27, claim 21, line 6.

In another embodiment of the present invention, a computer system for executing an object oriented program may comprise means for executing an object oriented program. Specification, page 10, lines 24-25; Specification, page 12, lines 3-23; Specification, page 29, claim 26, lines 1-2; Figure 1, elements 110, 122. The computer system may further comprise means for calling a first method from a first slot of a polymorphic inline cache. Specification, page 10, lines 25-26; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 15, line 19; Specification, page 29, claim 26, line 3; Figure 1, elements 110, 122. The computer system may further comprise means for calling a second method from a second slot of the polymorphic inline cache. Specification, page 10, lines 26-27; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 15, line 19; Specification, page 29, claim 26, lines 4-5; Figure 1, elements 110, 122. The computer system may further comprise means for determining if the first method and the second method have an identical object type. Specification, page 10, lines 27-28; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 26, lines 6-7; Figure 1, elements 110, 122. The computer system may further comprise means for calling the first method and the second method simultaneously if they do not have the identical object type. Specification, page 10, lines 28-30; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 26, lines 8-9; Figure 1, elements 110, 122. The computer system may further comprise means for preventing calling the second method from the second slot of the polymorphic inline cache until said means for

calling the first method has completed if they have the identical object type. Specification, page 10, line 30 – page 11, line 1; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 26, lines 10-12; Figure 1, elements 110, 122.

In one embodiment, the computer system as recited above, may further comprise means for invoking a first polymorphic inline cache initialisation routine. Specification, page 11, lines 3-4; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 28, lines 1-3; Figure 1, elements 110, 122. The computer system may further comprise means for locking the first slot of the polymorphic inline cache. Specification, page 11, lines 4-5; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 15, line 30; Specification, page 29, claim 28, line 4; Figure 1, elements 110, 122; Figure 2, step 210. The computer system may further comprise means for filling the first slot with a call instruction to the first method while the first slot is locked. Specification, page 11, lines 5-6; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 16, line 29; Specification, page 29, claim 28, lines 5-6; Figure 1, elements 110, 122. The computer system may further comprise means for updating the polymorphic inline cache so that a second method of a second type will invoke a second polymorphic inline cache initialisation routine. Specification, page 11, lines 6-8; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 28, lines 7-8; Figure 1, elements 110, 122. The computer system may further comprise means for making the first slot available to the first method. Specification, page 11, line 8; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 28, line 9; Figure 1, elements 110, 122. The computer system may further comprise means for calling the first method. Specification, page 11, lines 8-9; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 28, line 10; Figure 1, elements 110, 122. The computer system may

further comprise means for locking the second slot of the polymorphic inline cache. Specification, page 11, line 9; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 28, line 11; Figure 1, elements 110, 122. The computer system may further comprise means for filling the second slot with a call instruction to the second method while the second slot is locked. Specification, page 11, lines 9-11; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 28, lines 12-13; Figure 1, elements 110, 122. The computer system may further comprise means for updating the polymorphic inline cache so that an Nth method of an Nth type will invoke an Nth polymorphic inline cache initialization routine. Specification, page 11, lines 11-12; Specification, page 12, lines 3-23; Specification, page 14, line 30 – page 17, line 20; Specification, page 29, claim 28, lines 14-15; Figure 1, elements 110, 122.

VI. GROUND OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-5, 11-18 and 21-30 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Holzle et al. ("Optimizing Dynamically-Typed Object-Oriented Languages with Polymorphic Inline Caches") (hereinafter "Holzle") in view of Bacon (U.S. Patent No. 6,247,025). Claims 6-10 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon in further view of Harriman (U.S. Patent No. 6,182,177). Claims 19-20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon and in further view of Lee et al. ("Reducing Virtual Call Overheads in a Java VM Just-in-Time Compiler") (hereinafter "Lee").

VII. ARGUMENTA. Claims 1-5, 11-18 and 21-30 are not properly rejected under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon.

The Examiner has rejected claims 1-5, 11-18 and 21-30 under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon. Paper No. 6, page 2. Appellants respectfully traverse these rejections for at least the reasons stated below.

The reference, Bacon, which may qualify as prior art under 35 U.S.C. §102(e), does not preclude patentability under 35 U.S.C. §103 since Bacon and the claimed invention in claims 1-5, 11-18 and 21-30 were at the time the invention was made, subject to an obligation of assignment to the same person, which in this case was International Business Machines Corporation. 35 U.S.C. §103(c). Thus, Bacon is disqualified as being used as a prior art reference under 35 U.S.C. §103(c). Consequently, the rejections of claims 1-5, 11-18 and 21-30 are moot.

B. Claims 6-10 are not properly rejected under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon and in further view of Harriman.

The Examiner has rejected claims 6-10 under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon and in further view of Harriman. Paper No. 6, page 2. Appellants respectfully traverse these rejections for at least the reasons stated below.

The reference, Bacon, which may qualify as prior art under 35 U.S.C. §102(e), does not preclude patentability under 35 U.S.C. §103 since Bacon and the claimed invention in claims 6-10 were at the time the invention was made, subject to an obligation of assignment to the same person, which in this case was International Business Machines Corporation. 35 U.S.C. §103(c). Thus, Bacon is disqualified as being used as a prior art reference under 35 U.S.C. §103(c). Consequently, the rejections of claims 6-10 are moot.

- C. Claims 19-20 are not properly rejected under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon in further view of Lee and in further view of Shimura.

The Examiner has rejected claims 19-20 under 35 U.S.C. §103(a) as being unpatentable over Holzle in view of Bacon in further view of Lee and in further view of Shimura. Paper No. 6, page 2. Appellants respectfully traverse these rejections for at least the reasons stated below.

The reference, Bacon, which may qualify as prior art under 35 U.S.C. §102(e), does not preclude patentability under 35 U.S.C. §103 since Bacon and the claimed invention in claims 19-20 were at the time the invention was made, subject to an obligation of assignment to the same person, which in this case was International Business Machines Corporation. 35 U.S.C. §103(c). Thus, Bacon is disqualified as being used as a prior art reference under 35 U.S.C. §103(c). Consequently, the rejections of claims 29-20 are moot.

VIII. CONCLUSION

For the reasons noted above, the rejections of claims 1-30 are in error. Appellants respectfully request reversal of the rejections and allowance of claims 1-30.

Respectfully submitted,

WINSTEAD SECHREST & MINICK P.C.

Attorneys for Appellants

By: _____

Robert A. Voigt, Jr.

Reg. No. 47,159

Kelly K. Kordzik

Reg. No. 36,571

P.O. Box 50784
Dallas, Texas 75201
(512) 370-2832

APPENDIX

1. A method for implementing a polymorphic call site in a computer system executing an object oriented program, said method comprising the steps of:

a) creating a template of a polymorphic inline cache for a polymorphic call site, the template having a plurality of slots;

b) executing the polymorphic inline cache with an object of type k;

c) invoking a polymorphic inline cache initialisation routine;

d) finding an available k slot of the polymorphic inline cache;

e) locking the k slot of the polymorphic inline cache;

f) searching for a k method to call for the object of type k;

g) filling the k slot with a call instruction to the k method;

h) unlocking the k slot to complete the k slot; and

i) calling the k method of the object of type k.

2. The method of claim 1, wherein a first thread may initialize and/or access the k slot at the same time a second thread is initializing and/or accessing a slot other than the k slot.

3. The method of claim 2, wherein a second thread calling the k method while the first thread is initializing the k slot waits until after said unlocking step (h).

4. The method of claim 2, wherein a second thread calling the k method while the first thread is initializing the k slot searches for and calls the k method and leaves the polymorphic inline cache unchanged.

5. The method of claim 1, wherein a polymorphic inline cache is created for each polymorphic call site.

6. The method of claim 1, wherein said creating step (a) further comprises inserting a first illegal type value in a compare instruction of every slot of the polymorphic inline cache to indicate each slot is empty.

7. The method of claim 1, wherein said creating step (a) site further comprises putting a bit in each slot to indicate that each slot is empty.

8. The method of claim 6, wherein said locking step (e) further comprises replacing the first illegal type value in the compare instruction of the k slot with a second illegal type value to indicate the k slot is in use.

9. The method of claim 1, wherein said locking step (e) further comprises changing a bit in the k slot to indicate that the k slot is in use.

10. The method of claim 8, wherein said unlocking step (h) further comprises replacing the second illegal type value in the compare instruction of the k slot with a value of type k.

11. The method of claim 1, further comprising the step of (j) updating the polymorphic inline cache so that an object of type k+1 will initialize a corresponding k+1 slot.

12. The method of claim 11, wherein said updating step (j) further comprises inspecting the polymorphic inline cache to find the next empty slot.

13. The method of claim 12, wherein said polymorphic inline cache initialisation routine is the same for every object of the polymorphic inline cache.

14. The method of claim 11, wherein said updating step (j) further comprises

maintaining a state of the k slot or the k+1 slot.

15. The method of claim 14, wherein said polymorphic inline cache initialization routine is the same for every object of the polymorphic inline cache.

16. The method of claim 15, wherein said updating step (j) further comprises modifying the state of the k+1 slot.

17. The method of claim 1, wherein said invoking step (c) further comprises calling a different initialization routine for every object of a different type.

18. The method of claim 11, wherein said updating step (j) further comprises modifying the initialization routine so that upon a cache miss of the k slot, the k+1 initialization routine is called.

19. A system in a computing environment for implementing a polymorphic call sit, comprising:

- a central processing unit for executing an application;

- memory connected to the central processing unit via a bus;

- at least one input/output device connected to the bus and connected to a network interface to an external computer network;

- a just-in-time compiler for compiling object oriented applications for execution; and

- a polymorphic inline cache created by said just-in-time compiler, said polymorphic inline cache implementing a lockable slot for each individual object type to a polymorphic call site in the application.

20. A system in a computing environment for executing an application, comprising:

a processor for executing the application;
memory connected to the processor with an internal bus;
a network interface to connect to an external computer network; and
a polymorphic inline cache having a plurality of slots, each slot allocated to an object type of a method and locked to other object types of a polymorphic call site in the application.

21. A method of using a polymorphic inline cache, the method comprising the steps of:

calling a first method having a first object type from an executing object oriented program;

locking a first slot of the polymorphic inline cache with a call to the first method of the first object type; and

executing the first slot of the polymorphic inline cache.

22. The method of claim 21, further comprising the steps of:

calling a second method having a second object type;

locking a second slot of the polymorphic inline cache with a call to the second method of the second object type; and

executing the second slot of the polymorphic inline cache.

23. The method of claim 22, wherein the first slot of the polymorphic inline cache is executed simultaneously with the second slot of the polymorphic inline cache.

24. The method of claim 22, wherein the first method of the first object type is called from a first thread which executes independently from an executing second thread which called the second method of the second object type.

25. The method of claim 21, further comprising the steps of:

calling a second method having a second object type;
waiting until the first method of the first object type has executed;
determining whether the second object type and the first object type are the same object type; and
not locking a second slot of the polymorphic inline cache with a call to the method of the second type if the first and second object types are the same object type; and
locking the second slot with a call to the second method if the first and second object types are not the same object type and executing the second slot.

26. A computer system for executing an object oriented program, comprising:
means for executing an object oriented program;
means for calling a first method from a first slot of a polymorphic inline cache;
means for calling a second method from a second slot of the polymorphic inline cache;
means for determining if the first method and the second method have an identical object type;
means for calling the first method and the second method simultaneously if they do not have the identical object type; and
means for preventing calling the second method from the second slot of the polymorphic inline cache until said means for calling the first method has completed if they have the identical object type.

27. The computer system of claim 26, wherein the first method may be called from a first thread and the second method may be called from an independently executing second thread.

28. The computer system of claim 27, further comprising:
- means for invoking a first polymorphic inline cache initialisation routine;
 - means for locking the first slot of the polymorphic inline cache;
 - means for filling the first slot with a call instruction to the first method while the first slot is locked;
 - means for updating the polymorphic inline cache so that a second method of a second type will invoke a second polymorphic inline cache initialisation routine;
 - means for making the first slot available to the first method;
 - means for calling the first method;
 - means for locking the second slot of the polymorphic inline cache;
 - means for filling the second slot with a call instruction to the second method while the second slot is locked; and
 - means for updating the polymorphic inline cache so that a Nth method of a Nth type will invoke a Nth polymorphic inline cache initialization routine.

29. The method of claim 28, wherein the first, second, and Nth polymorphic inline cache initialization routines are identical.

30. The method of claim 28, wherein the first polymorphic inline cache initialisation routine is called from a first thread simultaneously while an independently executing second or Nth thread is calling the second or Nth method or is calling the second or Nth polymorphic inline cache initialization routine.



AF 2122 27W

PTO/SB/21 (09-04)

Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

18

Application Number

09/872,586

Filing Date

June 1, 2001

First Named Inventor

Stoodley et al.

Art Unit

2122

Examiner Name

Tang, Kuo Liang J.

Attorney Docket Number

CA920000035US1

ENCLOSURES (Check all that apply)

<input checked="" type="checkbox"/> Fee Transmittal Form	<input type="checkbox"/> Drawing(s)	<input type="checkbox"/> After Allowance Communication to TC
<input type="checkbox"/> Fee Attached	<input type="checkbox"/> Licensing-related Papers	<input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences
<input type="checkbox"/> Amendment/Reply	<input type="checkbox"/> Petition	<input checked="" type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief)
<input type="checkbox"/> After Final	<input type="checkbox"/> Petition to Convert to a Provisional Application	<input type="checkbox"/> Proprietary Information
<input type="checkbox"/> Affidavits/declaration(s)	<input type="checkbox"/> Power of Attorney, Revocation	<input type="checkbox"/> Status Letter
<input type="checkbox"/> Extension of Time Request	<input type="checkbox"/> Change of Correspondence Address	<input type="checkbox"/> Other Enclosure(s) (please identify below):
<input type="checkbox"/> Express Abandonment Request	<input type="checkbox"/> Terminal Disclaimer	
<input type="checkbox"/> Information Disclosure Statement	<input type="checkbox"/> Request for Refund	
<input type="checkbox"/> Certified Copy of Priority Document(s)	<input type="checkbox"/> CD, Number of CD(s) _____	
<input type="checkbox"/> Reply to Missing Parts/Incomplete Application	<input type="checkbox"/> Landscape Table on CD	
<input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53		

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name	Winstead Sechrest & Mirick P.C.		
Signature			
Printed name	Robert A. Voigt, Jr.		
Date	April 20, 2005	Reg. No.	47,159

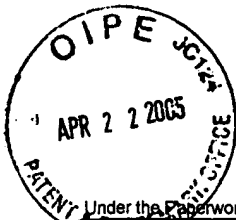
CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below:

Signature			
Typed or printed name	Serena Beller	Date	April 20, 2005

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.



PTO/SB/17 (11-04)

Approved for use through 07/31/2006. OMB 0651-0032

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

Effective on 10/01/2004. Patent fees are subject to annual revision.

FEE TRANSMITTAL

For FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27TOTAL AMOUNT OF PAYMENT (\$)**500.00****Complete if Known**

Application Number	09/872,586
Filing Date	June 1, 2001
First Named Inventor	Stoodley et al.
Examiner Name	Tang, Kuo Liang J.
Art Unit	2122
Attorney Docket No.	CA920000035US1

METHOD OF PAYMENT (check all that apply)☐ Check ☐ Credit Card ☐ Money Order☒ Deposit Account ☐ NoneDeposit
Account
Number
Deposit
Account
Name

09-0461

IBM Corporation

The Director is hereby authorized to: (check all that apply)

- ☒ Charge fee(s) indicated below
- ☐ Charge fee(s) indicated below, except for the filing fee
- ☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17
- ☒ Credit any overpayments

to the above-identified deposit account.

☐ Other (please identify):**WARNING:** Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**FEE CALCULATION****1. BASIC FILING FEE**

Fee Description	Fee (\$)	Small Entity Fee (\$)	Fee Paid(\$)
Utility Filing Fee	790	395	
Design Filing Fee	350	175	
Plant Filing Fee	550	275	
Reissue Filing Fee	790	395	
Provisional Filing Fee	160	80	

Subtotal (1) \$

FEE CALCULATION (continued)**2. EXTRA CLAIM FEES**

Fee Description	Fee (\$)	Small Entity Fee (\$)
Each claim over 20	50	25
Each independent claim over 3	200	100
Multiple dependent claims	360	180
For Reissues, each claim over 20 and more than in the original patent	50	25
For Reissues, each independent claim more than in the original patent	200	100

Total Claims	Extra Claims	Fee (\$)	Fee Paid (\$)
--------------	--------------	----------	---------------

- 20 or HP = _____ x _____ = _____
HP = highest number of total claims paid for, if greater than 20

Indep. Claims	Extra Claims	Fee (\$)	Fee Paid (\$)
---------------	--------------	----------	---------------

- 3 or HP = _____ x _____ = _____
HP = highest number of independent claims paid for, if greater than 3

Multiple Dependent Claims	Fee (\$)	Fee Paid (\$)
---------------------------	----------	---------------

Subtotal (2) \$

3. OTHER FEES

Fee Description	Fee (\$)	Small Entity Fee (\$)	Fee Paid(\$)
1-month extension of time	120	60	
2-month extension of time	450	225	
3-month extension of time	1,020	510	
4-month extension of time	1,590	795	
5-month extension of time	2,160	1,080	
Information disclosure stmt. fee	180	180	
37 CFR 1.17(q) processing fee	50	50	
Non-English specification	130	130	
Notice of Appeal	500	250	
Filing a brief in support of appeal	500	250	500.00
Request for oral hearing	1,000	500	
Other:			

Subtotal (3) \$ 500.00

SUBMITTED BY

Signature

Name (Print/Type)

Robert A. Voigt, Jr.

Registration No.
(Attorney/Agent)

47.159

Telephone 512.370.2832

Date April 20, 2005

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.